**Computer Science and Application** 计算机科学与应用, 2022, 12(4), 1139-1148
Published Online April 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.124116

Hans汉斯

# 基于区块链的个人敏感数据安全存储及共享方法

**曹　穗**

广东工业大学，计算机学院，广东 广州

收稿日期：2022年3月18日；录用日期：2022年4月19日；发布日期：2022年4月26日

## 摘　要

**针对云环境中的数据安全共享困难、隐私信息易泄露和加解密开销大的问题，本文提出一种基于区块链的个人敏感数据安全存储及共享方法。方案利用以太坊区块链平台搭建实验环境，采用改进的国密SM2数字签名算法对敏感数据加以保护，部署在区块链上的智能合约能够执行自动化的属性判断，实现了中心化的访问控制机制。用户的访问记录都保存在区块链中，保证可溯源追责。实验分析表明，该方案在数据安全性、隐私保护等方面有明显的优势。**

## 关键词

**区块链，SM2数字签名，隐私保护，数据安全共享**

# Secure Storage and Sharing Method of Personal Sensitive Data Based on Blockchain

**Sui Cao**

School of Computer, Guangdong University of Technology, Guangzhou Guangdong

Received: Mar. 18th, 2022; accepted: Apr. 19th, 2022; published: Apr. 26th, 2022

## Abstract

**Aiming at the difficulties of data security sharing in the cloud environment, the easy leakage of private information, and the high overhead of encryption and decryption, this paper proposes a blockchain-based method for secure storage and sharing of personal sensitive data. This scheme uses the characteristics of blockchain to build an experimental platform, and at the same time uses the improved national secret SM2 digital signature algorithm to protect sensitive data. The**

文章引用: 曹穗. 基于区块链的个人敏感数据安全存储及共享方法[J]. 计算机科学与应用, 2022, 12(4): 1139-1148.
DOI: 10.12677/csa.2022.124116

**smart contracts deployed on the blockchain can perform automatic attribute judgment and realize centralized access control. The user's access records are stored in the blockchain to ensure traceability and accountability. Experimental analysis shows that this scheme has obvious advantages in data security, privacy protection, etc.**

## Keywords

**Blockchain, Chinese SM2 Digital Signature Algorithm, Privacy Protection, Data Security Sharing**



## 1. 引言

在信息化时代，人们常常需要使用到类似出生证明、健康证明、财产契约和学术成绩单等的官方文件，此类数据由一些公认的可信机构发布，或在研究机构中贡献研究价值，例如购物记录、医疗数据等；或由所有者在与其他人或组织的交涉中使用，以证明某些陈述的真实性，例如健康证明、财产证明、学籍信息、征信数据等。这类数据往往包含有用户隐私信息，而用户习惯于将个人数据存储在半可信的云服务商中，若将数据明文存储在云上，用户无法控制云平台上对数据的使用，易于造成了个人隐私数据的泄露风险；若将数据密文存储在云上，数据加解密计算耗费巨大，难以实现高效共享。为了使这类数据能更好地在研究或特定场景下发挥其价值，加密数据云存储的访问控制是非常重要的，因此迫切需要一个既能有效保护用户隐私信息，又能高效实现数据共享的方案。

自 2008 年比特币概念被中本聪提出，经过十多年的发展，作为比特币底层技术的区块链也逐渐被学者探索其在数字货币领域外的应用。尤其在信息安全领域，区块链以其去中心化、公开透明、不可篡改等特性，获得诸多学者在解决隐私数据安全共享问题上的青睐。ZYSKIND [1]等提出基于一个去中心化用户数据管理框架，但该系统要求数据所有者与数据访问者必须同时在线；Jemel [2]等提出了一种具有时间维度的分布式访问控制模型，该方案结合区块链技术与 CP-ABE 算法来验证用户的访问权限，但该方案无法满足复杂的访问需求；Li [3]等提出了基于区块链的分布式基于多授权 CP-ABE 和 DMA-ABS 的数据存储与共享系统，数据所有者可以安全地与多个满足策略的用户共享数据，不需要单独授予单独的权限，但数据所有者无法动态更改数据访问策略；SHU [4]等基于多门限哈希函数构造了一种 MCPS 无证书聚合签名方案，实现基于区块链的医疗信息的安全存储和共享，具有较高的计算效率和存储效率，但方案中没有实现患者对医疗数据的自主控制权；Gao [5]等提出了一种优化的基于区块链的策略隐藏方案 OHP-CP-ABE，并使用乘法同态 ElGamal 密码系统来确保授权验证期间的属性隐私，保证了策略和属性隐私的同时实现可信访问，但双线性映射和指数运算的频繁运用使得加重算法计算开销；Niu [6]等采用可搜索加密技术实现区块链上的安全搜索，利用区块链的不可否认性确保关键字和密文的安全性，验证算法保证了云上数据的完整性，但该方案的属性撤销由权威中心负责，用户对数据未能实现完全控制；文献[7]提出了基于以太坊平台的访问控制方案，方案设计了访问控制、判断和注册三大合约，访问者的属性判断、授权以及记录存档都由智能合约自动化完成。文献[8]实现了分布式物联网设备配置文件的管理，智能合约记录物联网设备的访问控制操作，配置文件和访问数据形成共识后上传到私有链中，私链的运用保证了较高的隐私性，但中心化程度也相对较高。

从上述分析来看，区块链技术应用于隐私数据保护领域方兴未艾，且大多数学者倾向于设计复杂的访问控制策略来实现对数据的保护，实际上，直接把隐私数据保护起来也许能更加简单、安全、有效地达到期望的效果。因此本文提出一种基于区块链的个人敏感数据安全存储及共享方法，利用区块链的特性搭建去中心化的访问控制平台，编写智能合约执行访问者属性自动判断，在此基础上，使用改进的国密算法 SM2 隐藏用户数据中的敏感信息，提高整体安全性和性能，融合的内容提取签名的思想实现了无需经过数据发布方的确认的情况下验证被提取后的数据的真实性。

## 2. 相关知识

### 2.1. 区块链

区块链[9]本质上是一个去中心化的分布式账本数据库，每个区块以按时间先后顺序链接，后产生的区块指向前一个区块，当前区块的区块头包含上一个区块的哈希值。当链中某个区块数据被篡改，则该区块的哈希值发生变化，在此区块的所有区块都会发生变化。一个完整的区块链系统由密码学，P2P 网络和工作量证明等多种技术组成，P2P 网络组成了区块链底层的分布式网络系统；时间戳、数字签名保证数据库的不可篡改和可追溯；拜占庭容错机制和共识算法保证节点间区块数据的一致性。

### 2.2. 智能合约

智能合约[10]的概念最早于 1994 年由尼克·萨博提出。由于当时缺少能够支持可编程合约的数字系统和技术，关于智能合约的工作理论迟迟没有推进。直到比特币的诞生，学者们认识到其底层技术区块链能为智能合约提供可信执行环境，智能合约自有了一个可支持的平台。智能合约是用户自定义的模块化的、可重用的、部署在区块链上的脚本，一旦满足预设条件，智能合约就可以在没有第三方介入的情况下自动执行。通过使用智能合约，可以实现可信的交易，并且这些交易是可追踪和不可逆的。

### 2.3. 内容截取签名

内容截取签名[11] (Content- Extraction Signatures, CES)与传统标准数字签名不同之处在于，在多方参与的情景下，允许消息签名的持有者在不与消息原始签名者进行交互的情况下，提取原消息的一部分，并为这部分内容计算一个可公开验证的签名，签名验证者无需和原始签名者交互即可确认被截取后数据的真实性。

在内容截取签名中，消息 $M$ 被看作是 $n$ 个子消息段组成的集合，即 $M=\{m_1,m_2,\cdots,m_n\}$，定义子消息编号集合 $I=\{1,2,\cdots,n\}$，被提取的子消息集合记为 $M'$，$M'$ 中的子消息的编号记作截取子集 $CI(M')$，$CI(M')\in I$。签名者设置内容截取访问结构 $CEAS\subseteq I$，满足 $CEAS\subseteq CI(M')$ 时截取方式合法，否则判定为非法截取。

### 2.4. SM2 签名算法

SM2 算法[12]是我国自主研发的一种基于椭圆曲线离散对数困难问题的公钥密码算法，包含了数字签名算法、密钥交换协议和公钥加密算法。其中，SM2 签名算法相比于 RSA 签名算法，其密码复杂度高、处理速度快、机器性能消耗更小，可以很好地应用于本方案的敏感数据签名环节。SM2 数字签名算法如下：

1) 密钥产生：输入一个有限域 $F_p$ 上椭圆曲线参数集合，用产生随机数 $d\in[1,n-1]$，计算 SM2 椭圆曲线上基点 $G$ 的 $d$ 倍点，记为 $P=[d]G$，则 $P$ 为公钥，$d$ 为私钥。

2) 签名：输入椭圆曲线参数、签名者私钥及消息 $M$

①计算杂凑值 $Z_A = H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$， $ID_A$ 是签名者 A 长度为 $ENTL_A$ 的可辨别标识；

②计算 $M$ 消息摘要 $e = H_{256}(Z_A \| M)$，$M$ 为待签名消息；

③产生随机数 $k \in [1, n-1]$，计算椭圆曲线点 $(x_1, y_1) = [k]G$；

④计算参数 $r = (e + x_1) mod\ n$，参数 $s = \left((1+d)^{-1} \cdot (k - r \cdot d)\right) mod\ n$；

⑤输出消息 $M$ 的签名为 $(r, s)$；

3) 签名验证：输入椭圆曲线参数、签名者公钥 $P$、待验证消息 $M'$ 以及由签名者传递而来的 $M'$ 的签名 $(r', s')$

①计算 $M'$ 消息摘要 $e' = H_{256}(Z_A \| M')$，计算 $t = (r' + s') mod\ n$，若 $t = 0$，验证失败；

②计算椭圆曲线点 $(x_1', y_1') = [s']G + [t]P$；

③若 $r' = (e' + x_1') mod\ n$，验证通过；否则签名无效。

## 3. 论基于区块链的个人敏感数据安全存储及共享方法

### 3.1. 方案构造

本文提出的方案如图 1 所示，该模型主要由数据生成者(Data Generator, DG)数据持有者(Data Owner, DO)、数据访问者(Data Visitors, DV)、云服务提供商(Cloud Service Provide, CSP)、和区块链共识网络组成。数据生成者是创建一类官方数据的可信机构，在本方案的签名过程中承担签名者角色；数据持有者拥有数据所属权，在某些场景下，可以通过数据生成者为其创建的数据证明自身某些资质或发挥其研究价值，但当该文件中包含有涉及其隐私信息时，数据持有者希望可以在不泄露隐私信息的前提下将数据共享给数据访问者；区块链模块作为系统的中心，负责用户处理请求，数据传递以及访问控制；数据请求者向系统请求数据访问权限，并在签名过程中承担验证者角色；云服务提供商提供数据存储服务，避免海量数据堆积在区块链上。

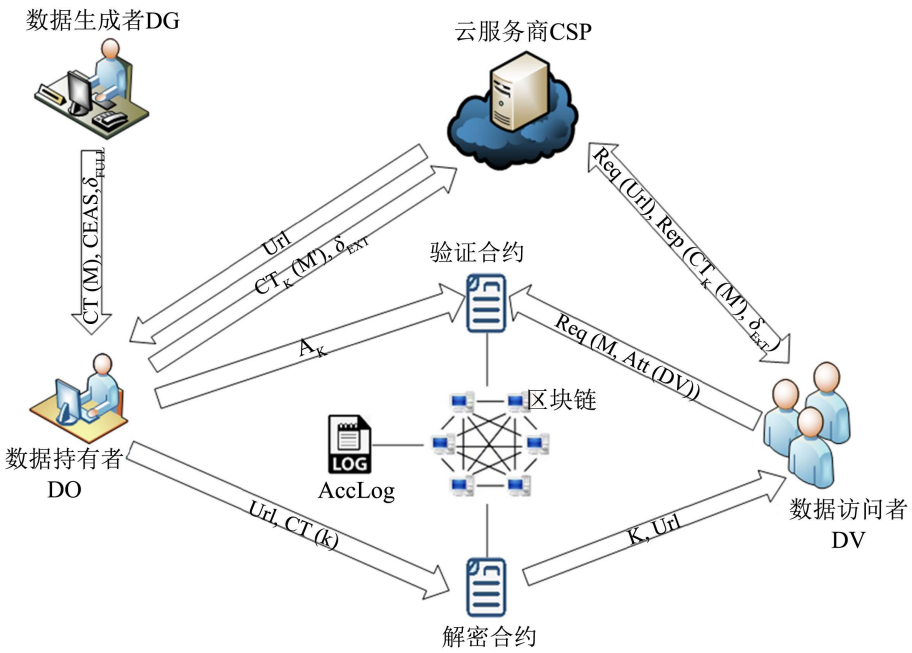


**Figure 1.** Diagram of the scheme overall structure
**图 1.** 方案总体构造图

以下是本方案主要步骤的：

1) 初始化：$Setup(\lambda)\to(params)$，输入安全参数 $\lambda$，定义 SM2 椭圆曲线 $E$，选取阿贝尔群，确定椭圆曲线的基点 $G$，以及密码杂凑函数 $Hv$，输出系统公开参数 $params$。

2) 秘钥对生成：$KeyGen(att_1,att_2,\cdots,att_n,params)\to(pk_i,sk_i)$，所有用户分别注册以太坊外部账户 EOA 并生成关联用户属性的 SM2 算法密钥对。

3) 数据预处理：DG 生成数据 $M$，DG 定义 $M$ 的内容截取访问结构 $CEAS$，并将 $M$ 划分为 $n$ 个子消息，计算杂凑值 $Z_{DG}=Hv(ENTLN_{DG},ID_{DG},params)$。

4) 全局签名：$Sign(Z_{DG},sk_{DG},Hv(M),CEAS)\to\delta_{FULL}$，用 DG 的私钥生成消息 $M$ 的全局签名 $\delta_{FULL}$。

5) 消息传递：DG 用 DO 的公钥加密消息 $M$，将 $M$ 秘文，哈希函数，加密方法，全局签名通过安全信道发送给 DO。

5) 提取签名 $Sign(Z_{DG},sk_{DG},Hv(M),CEAS)\to Ext(M,CI(M'),CEAS,\delta_{FULL})\to\delta_{EXT}$，DO 接收来自 DG 的消息 $M$、全局签名 $\delta_{FULL}$ 及 $CEAS$；DO 验证 $\delta_{FULL}$ 的正确性，出于隐私保护需求对 $M$ 提取一部分内容并生成提取签名 $\delta_{EXT}$。

6) 数据上传：DO 加密提取的消息 $M'$，并将 $M'$ 的密文 $Ek(M')$ 和签名 $\delta_{EXT}$ 上传至云服务商，检索路径为 $url$。

7) 设置合约：DO 将 $M'$ 的密钥 $k$ 的访问策略 $A_k$ 写入验证合约，将 $k$ 的密文 $C_k$ 和路径 $url$ 写入解密合约。

8) 访问数据：DV 向区块链访问控制中心发起请求，验证合约检验其身份属性，若符合访问结构，则触发解密合约解密 $C_k$ 并将 $k$ 和 $url$ 返回给 DV。

9) 数据上链：主节点验证并收集访问池中的合法请求，并向所有共识节点广播。当超过 51%的共识节点形成共识，DV 的访问记录将被写入块中，新创建的区块接入到区块链。

10) $Verify(M',ID_A,\delta_{EXT},P_A)\to\{0,1\}$：DV 解密从云端下载的 $M'$ 密文，验证消息 $M'$ 的签名 $\delta_{EXT}$ 正确性。

### 3.2. 具体实施

1) 初始化

$Setup(\lambda)\to(params)$：输入安全参数 $\lambda$，输出系统公共参数 $params=\{p,F_q,a,b,n,G,Hv\}$。确定有限域 $Fq$ 上非奇异 SM2 椭圆曲线 $E:y^2=x^3+ax+b\ mod\ p(a,b\in F_q)$，其中 $p$ 为大素数，在包含无穷远点和 E 的所有点中选取循环群 $\mathbb{G}$，$n$ 阶基点 $G=(x_G,y_G)(G\neq O)$，安全的哈希函数 $Hv:\{0,1\}^*\to Z_p^*$。

2) 秘钥对生成

$KeyGen(att_1,att_2,\cdots,att_n,params)\to(pk_i,sk_i)$：用户注册以太坊外部账户，输入用户身份信息，根据用户属性 attributes 和 SM2 签名算法的秘钥生成规则产生用户密钥对。以签名者 A 的密钥对生成为例：随机选择整数 $d\in Z_q^*$，计算 A 的 SM2 签名密钥对 $(d_A,P_A)=(d,x_A,y_A)=(d,[d]G)$，$d_A$ 为私钥，由 A 秘密保存，$P_A$ 为公钥。

3) 数据预处理

记签名者 A 长度为 $entlen_A$ 比特的可辨别标识为 $ID_A$，记 $ENTLA$ 为由整数 $entlen_A$ 转换而成的两个字节。A 创建数据 $M$，并将 $M$ 划分为 $j$ 个子消息，记为 $M=\{M[1],M[2],M[3],\cdots,M[j]\}$，子消息编号集合 $I=\{1,2,3,\cdots,j\}$。A 对 $M$ 设置内容截取访问结构 $CEAS\in I$，编号被包含于 $CEAS$ 的子消息都为必选消息，否则非法。

4) 生成签名

签名者 A 计算杂凑值 $Z_A = H_{256}\left(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A \| CEAS\right)$，

$Sign\left(Z_{DG}, d_A, H_{256}(M), CEAS\right) \to \delta_{FULL}$，签名者 A 用秘钥 $d_A$ 生成 $M$ 的全局签名 $\delta_{FULL}$：

①A 随机选取 $k \in [1, n-1]$，计算椭圆曲线点 $(x_1, y_1) = [k]G$，将 $x_1$ 的数据类型转换为整数；

②对每个 $i \in I$ 的子消息 $M[i]$，计算 $h[i] = H_{256}\left(M[i] \| CEAS \| T \| Z_A\right)(i \in [1, j])$，其中 $T$ 为长度固定的 *CEAS* 标记 *CEAS-Tags*，在签名中随机选取；

③将 $h[i]$ 的数据类型转换为整型，计算 $r[i] = \left(h[i] + x_1\right) mod\ n$，$s[i] = \left((1+d)^{-1}\left(k - r[i]d\right)\right) mod\ n$，若 $s[i] = 0$ 则重新选取随机数 $k$；

④计算 $r = \prod_{i \in [1, j]} r[i] mod\ n$，$s = \prod_{i\, i \in [1, j]} r[i] mod\ n$；

⑤将 $r$、$s$ 的数据类型转换为字节串，消息 $M$ 的全局签名为 $\delta_{FULL} = \left(CEAS \| T \| r \| s\right)$。

5) 消息传递

A 使用数据持有者 DO 公钥 *Pko* 加密消息 $M$，并将 $M$ 密文、哈希函数、加密方法，全局签名通过安全信道发送给 DO。

6) 提取签名

$Ext\left(M, CI(M), CEAS, \delta_{FULL}\right) \to \delta_{EXT}$：DO 用其私钥 *SKo* 解密出 $M$ 明文，用 A 的公钥 $P_A$ 验证 $M$ 全局签名 $\delta_{FULL}$ 的正确性，验证方法即 SM2 标准验签过程[13] (参见《SM2 椭圆曲线公钥密码算法》)，根据需要针对消息 $M$ 的提取一个可验证的签名：

DO 设置截取子集 $CI(M') \in I$，$X \in CI(M')$，截取消息 $M' = \left\{M[i] \mid i \in X\right\}$，截取签名 $\delta_{EXT} = \left(CEAS \| T \| r'[i]_{i \in X} \| s'[i]_{i \in X} \| CI(M')\right)$。

7) 数据上传

DO 用对称算法加密截取数据 $M'$，对称算法秘钥为 key，则 $M'$ 密文记为 $CT(M') = ENCkey(M')$，$CI(M')$ 及截取签名 $\delta_{EXT}$ 上传到云存储平台，云平台返回路径为 *url*。

8) 部署智能合约

DO 将密钥 key 用基于密文策略的属性加密算法加密，key 的访问策略 $A_k$ 添加至验证合约，将 key 的密文 $CT(key)$ 和路径 *url* 上传至解密合约。

9) 数据访问

数据访问者 DV 登录以太坊账户，向区块链中心请求对数据 $M$ 的访问；区块链中心处理访问者请求并触发合约运行，验证合约调动 DO 添加的访问策略 $A_k$ 与访问者 DV 的属性进行比对，一旦满足 $Att(DV) \in A_k$，解密合约自动执行解密属性基加密算法并把 $k$ 和 *url* 传递给 DV。

10) 数据上链

区块链网络主节点像所有共识节点广播访问者 DV 的访问记录，达成共识后，DV 的访问记录接入到新创建的区块区块链，形成访问日志 *AccLog*。

11) 签名验证

访问者根据路径 *url* 获取存储在云端的密文 $CI(M')$ 和截取签名 $\delta_{EXT}$，用对称密钥 $k$ 解密得到截取数据的明文 $M'$ 后，根据 DO 的公钥 $P_A$ 验证截取签名的正确性。

$Verify\left(M', \delta_{EXT}, ID_A, P_A, params\right) \to \{0,1\}$，输入 $M'$ 明文、$\delta_{EXT}$、签名者 A 身份标识 $ID_A$、公钥 $P_A$ 以及公共参数 *params*；若输出为 1，表示验证通过；若输出为 0，算法终止，验证失败。验证算法过程见下：

①验 $CEAS \subseteq CI(M')$ 是否成立，若成立进行下一步，否则返回 0；

②检验 $r''[i]$、$s''[i] \in [1, n-1]$ 是否成立，若成立进行下一步，否则返回 0；

③对每个编号 $i \in CI(M')$ 的子消息，计算 $h'[i]=H_{256}\left(M'[i]_{i\in CI(M')} \| CEAS \| T \| Z_A\right)$，将 $h'[i]$ 的数据类型转换为整数；

④将 $r''[i]$、$s''[i]$ 转为整型，计算 $t[i]=\left(r''[i]+s''[i]\right) mod\ n$，若 $t[i]\neq 0$，进行下一行，否则返回 0；

⑤计算椭圆曲线点 $(x_1', y_1')=s''[i]G+t[i]P_A$；

⑥将 $x_1'$ 的数据类型转换为整数，若 $r''[i]=\left(h'[i]+x'\right) mod\ n$ 成立，则签名验证通过。

## 4. 方案分析

### 4.1. 正确性分析

本方案在密码算法层面主要用到的是改进的 SM2 数字签名算法，该算法的正确性体现在等式 $r''[i]=\left(h'[i]+x_1'\right) mod\ n$ 的成立，只要确保该等式成立，即可保证被提取后的数据未经篡改，签名来源依然属于数据签名者 A，且在验证过程中验证者无需与签名者交互。验证过程：

$$\begin{aligned}(x_1', y_1') &= s''[i]G+t[i]P_A \\ &= s''[i]G+\left(r''[i]+s''[i]\right)P_A \\ &= s''[i]G+s''[i][d_A]G+r''[i]P_A \\ &= \left(1+d_A\right)s''[i]G+r''[i]P_A\end{aligned}$$

由签名过程 $s[i]=\left(\left(1+d\right)^{-1}\left(k-r[i]d\right)\right) mod\ n$ 知 $\left(1+d\right)=\left(k-r[i]d\right)s[i]^{-1}$ 若签名来源确属于签名者 A，则 $d_A=d$，代入上式可得：$(x_1', y_1')=\left(k-r[i]\cdot d\right)s[i]^{-1}\cdot s''[i]G+r''[i]P_A$

假设 DO 是诚实的，那么验证者收到截取签名 $\delta_{EXT}$ 中的的 $r''[i]$、$s''[i]$ 应与 $s[i]$、$r[i]$ 相等，可得：

$$(x_1', y_1')=\left(k-r[i]\cdot d\right)G+r[i]P_A=kG=(x_1, y_1)$$

得出 $x_1'=x_1$；

因此

$$r''[i]=\left(h'[i]+x_1'\right) mod\ n=\left(h'[i]+x_1\right) mod\ n=r[i]=\left(h[i]+x_1\right) mod\ n,$$

得出：$h'[i]=h[i]_{i\in CI(M')}$。

DV 计算得到的消息摘要 $h'[i]=h[i]$，即证明消息 $M'$ 未经篡改。等式验证通过，方案正确。

### 4.2. 安全性分析

#### 4.2.1. 隐私保护分析

首先，用户注册以太坊账户，鉴于区块链匿名性的特点，所有操作都是通过以太坊区块链上的地址发送的，该地址独立于用户的个人信息，不能将区块链上的数据和个人相关联，一定程度上降低了身份隐私泄露的风险。其次，数据持有者在符合签名者设置的内容截取访问结构前提下，对数据的使用拥有完全控制权，数据以密文形式存储在半可信云存储服务商 CSP，CSP 无法获得数据铭明文也即避免用户数据被滥用。同时的，为加密数据的对称秘钥设置的访问策略通过基于属性的加密算法保护起来，实现了一对多的细粒度访问控制，算法逻辑作为预设条款被写入智能合约，智能合约的执行原理有效降低未授权节点或恶意节点获得数据访问权限的可能性。

#### 4.2.2. 数据安全分析

假设不诚实的 DO 试图在提取签名过程中改动任意子消息 $M[i]$，或打乱子消息顺序引起语义混乱，都会导致验证者在计算子消息摘要时 $h'[i]\neq h[i]$，使得等式 $r''[i]=\left(h'[i]+x_1'\right) mod\ n$ 不成立，验证失败。

为避免数据被恶意提取导致丧失其基本价值，算法引入截取规则 *CEAS*，编号属于 *CEAS* 的子消息都为必选消息，当 $CEAS \subseteq CI(M')$ 时截取方式合法。假设不诚实的 DO 试图在提取签名过程中非法截取签名，验证签名时首先检验出 $CEAS \subseteq CI(M')$ 不成立，验证失败。

在本方案数字签名算法中，未被提取的子消息对于验证者是不可见的，攻击者想要获取被隐藏的消息是不可行的。

假设不诚实的 DO 或恶意攻击者试图改动数据 *M* 后伪装成 DG 生成一份原始签名，但由于数据传递过程中，在只知 DG 公钥情况下，DO 或恶意攻击者难以突破椭圆曲线离散对数难解性计算出 DG 密钥。

且 DG 的身份信息 $ID_A$ 早已聚合在所对应的杂凑值 $Z_A$ 中，不是用户 A 所对应的杂凑值，验证自然通不过。

### 4.3. 功能分析

下表 1 为本文方案与文献[1] [2] [7] [14]的功能分析对照，分析表明，要同时满足去中心化、访问控制 秘密隐藏、不可伪造与追踪溯源这五大性能是充满挑战性的。由表可知，所有的方案均实现了去中心化的访问控制，这也是区块链应用于数据安全共享领域的一大重要标准，文献[1] [7]在秘密隐藏和不可伪造性方面缺少具体描述；文献[2] [14]则未能实现秘密隐藏和访问记录追踪溯源，相比之下，本文方案在性能均衡上略有优势。

**Table 1.** Program functional analysis and comparison
**表 1.** 方案功能性分析对比

| 方案 | 去中心化 | 细粒度的访问控制 | 秘密隐藏 | 不可伪造性 | 追踪溯源 |
|---|---|---|---|---|---|
| 文献[1] | √ | √ | × | × | √ |
| 文献[2] | √ | √ | × | √ | × |
| 文献[7] | √ | √ | × | × | √ |
| 文献[14] | √ | √ | × | √ | × |
| 本文方案 | √ | √ | √ | √ | √ |

## 5. 实验与结果分析

### 5.1. 实验环境

本文实验硬件环境为 Intel (R) Core (TM) i5-8500 CPU @3.40 GHz，8 GB RAM；操作系统为 64 位 Windows 10，使用在线版 Remix IDE 进行智能合约的开发和测试，智能合约部署在以太坊区块链上，编程语言为 Java 和 Solidity；外部资源库为 JPBC 和 web3.js，其中 JPBC 是常用于基于配对的密码学算法仿真的 Java 封装库，web3.js 工具包包含了以太坊提供的一系列与区块链交互的 Javascript 对象和函数以及与智能合约交互的 API。

### 5.2. 实验结果分析

在本文方案中，实现细粒度的访问控制采用的加密算法是基于密文的略的属性加密，ABE 基于双线性对实现，双线性对操作成本昂贵，但本文中 CP-ABE 的加密对象是对称加密密钥而非数据明文，数据明文采用对称加密算法加密，这个方法大大减轻了加解密的开销。

同时的，在签名算法的选择上，采用了改进的国密 SM2 数字签名算法。SM2 算法是基于椭圆曲线上

点群离散对数难题的非对称加密算法，相比于 RSA 的安全性依赖于大整数的分解困难性，256 位的 SM2 密码强度已经比 2048 位的 RSA 密码强度要高，且由于目前所知求解 ECDLP 的最好方法是指数级的，这使得我们选用 SM2 算法作数字签名时，所要求的密钥长度比 RSA 要短得多。

通过统计方案中所使用到的运算，对比本文提出的方案与文献[15] [16]中的方案的进行性能分析，如下表 2 所示，其中 $n$ 表示消息被 $M$ 划分成子消息的数目，$m$ 表示截取子集 $CI(M')$ 中子消息的数目；*par* 表示双线性对运算，*exp* 表示幂运算，*sca* 表示基于椭圆曲线密码编码的标量乘法运算，*hash* 表示消息摘要函数运算。通过对比可知，方案[15]和[16]采用了相对耗时的幂运算和双线性对运算，本文采用的是基于椭圆曲线的标量乘法运算，在相同的安全级别上，在计算效率方面具有优势。

**Table 2.** Computational cost analysis
**表 2.** 计算开销分析

| 方案 | 签名算法 | 验证算法 |
|---|---|---|
| 文献[15] | $n$ *hash* + 2 *exp* | $m$ *hash* + 3 *exp* |
| 文献[16] | 1 *hash* + 2 *par* + 2 *sca* | 1 *hash* + 1 *par* + 3 *sca* |
| 本文方案 | (1 + $n$) *hash* + *sca* | $m$ *hash* + 2 *sca* |

## 6. 总结与展望

本文通过以太坊提供的平台，结合改造的 SM2 签名算法构造了一个基于区块链技术的个人敏感数据安全共享机制，本方案专注于用户数据的机密性、不可伪造性，实现了数据的隐私保护与安全共享的有效平衡。未来针对区块链平台确认交易的时间长，吞吐量不高而导致在部分云存储场景中制约整个系统效率的问题，考虑针对特定场景通过改进共识算法以缩短共识时间。

## 参考文献


[1] Zyskind, G., Nathan, O. and Pentland, A.S. (2015) Decentralizing Privacy: Using Blockchain to Protect Personal Data. 2015 *IEEE Security and Privacy Workshops*, San Jose, 21-22 May 2015, 180-184. https://doi.org/10.1109/SPW.2015.27

[2] Jemel, M. and Serhrouchni, A. (2017) Decentralized Access Control Mechanism with Temporal Dimension Based on Blockchain. 2017 *IEEE* 14*th International Conference on e-Business Engineering* (*ICEBE*), Shanghai, 4-6 November 2017, 177-182. https://doi.org/10.1109/ICEBE.2017.35

[3] Li, G. and Sato, H. (2019) A Privacy-Preserving and Fully Decentralized Storage and Sharing System on Blockchain. 2019 *IEEE* 43*rd Annual Computer Software and Applications Conference* (*COMPSAC*), Milwaukee, 15-19 July 2019, 694-699. https://doi.org/10.1109/COMPSAC.2019.10289

[4] Shu, H., Qi, P., Huang, Y., Chen, F., Xie, D. and Sun, L. (2020) An Efficient Certificateless Aggregate Signature Scheme for Blockchain-Based Medical Cyber Physical Systems. *Sensors*, **20**, Article No. 1521. https://doi.org/10.3390/s20051521

[5] Gao, S., Piao, G., Zhu, J., *et al*. (2020) TrustAccess: A Trustworthy Secure Ciphertext-Policy and Attribute Hiding Access Control Scheme Based on Blockchain. *IEEE Transactions on Vehicular Technology*, **69**, 5784-5798. https://doi.org/10.1109/TVT.2020.2967099

[6] Niu, S., Chen, L. and Liu, W. (2020) Attribute-Based Keyword Search Encryption Scheme with Verifiable Ciphertext via Blockchains. 2020 *IEEE* 9*th Joint International Information Technology and Artificial Intelligence Conference* (*ITAIC*), Chongqing, 11-13 December 2020, 849-853. https://doi.org/10.1109/ITAIC49862.2020.9338962

[7] Zhang, Y., Kasahara, S., Shen, Y., *et al*. (2018) Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet of Things Journal*, **6**, 1594-1605. https://doi.org/10.1109/JIOT.2018.2847705

[8] Košťál, K., Helebrandt, P., Belluš, M., *et al*. (2019) Management and Monitoring of IoT Devices Using Blockchain. *Sensors*, **19**, Article No. 856. https://doi.org/10.3390/s19040856

[9] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. Decentralized Business Review.

[10] Szabo, N. (1996) Smart Contracts: Building Blocks for Digital Markets. *Extropy*: *The Journal of Transhumanist Thought*, **18**, 28.

[11] Steinfeld, R., Bull, L. and Zheng, Y. (2001) Content Extraction Signatures. *International Conference on Information Security and Cryptology*, Seoul, 6-7 December 2001, 285-304. https://doi.org/10.1007/3-540-45861-1_22

[12] 汪朝晖, 张振峰. SM2 椭圆曲线公钥密码算法综述[J]. 信息安全研究, 2016, 2(11): 972-982.

[13] 国家密码管理局. SM2 椭圆曲线公钥密码算法[EB/OL]. http://www.sca.gov.cn/sca/xwdt/2010-12/17/content_1002386.shtml, 2010-12-17.

[14] Huang, H.P., Zhu, P., Xiao, F., *et al*. (2020) A Blockchain-Based Scheme for Privacy-Preserving and Secure Sharing of Medical Data. *Computers & Security*, **99**, Article ID: 102010. https://doi.org/10.1016/j.cose.2020.102010

[15] 王彩芬, 徐婷, 张玉磊, 杨小东. 基于可截取签名和属性加密的云存储访问控制方案[J]. 计算机工程与科学, 2015, 37(2): 238-244.

[16] Wang, M., Zhang, Y., Ma, J., *et al*. (2020) A Universal Designated Multi Verifiers Content Extraction Signature Scheme. *International Journal of Computational Science and Engineering*, **21**, 49-59. https://doi.org/10.1504/IJCSE.2020.10026865